ML Algorithms Explained:

Supervised ML algorithms:

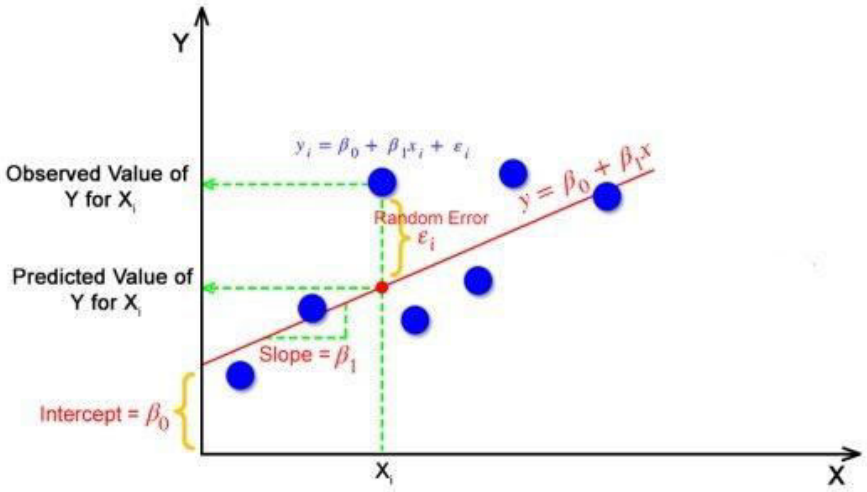
1. Linear Regression
2. KNN
3. Naïve Bayes
4. Decision trees
5. Random Forest
6. Support Vector Machines
7. Neural Networks

Unsupervised ML algorithms:

1. K Means Clustering
2. Association rules

==========

1. Linear Regression: (https://www.youtube.com/watch?v=CtsRRUddV2s)



Helps to solve supervised ml problems.

Y = mx + c for single feature

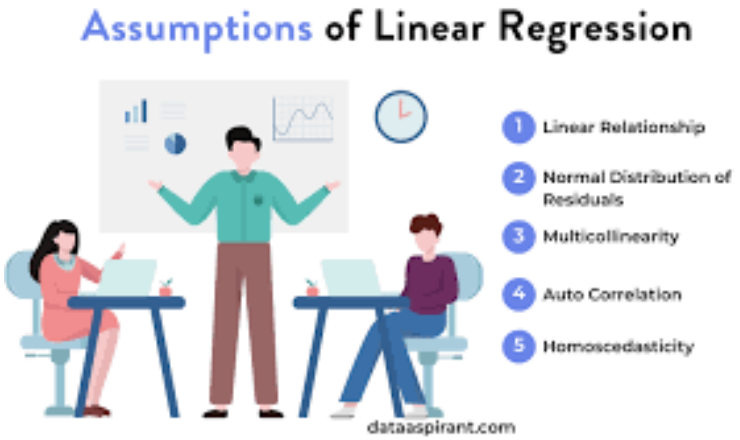
Where m is slope and c is the intercept of the fitted line.

Y = b0 + b1x + b2x + ...bnx (for b0 to bn features)

Used Case: Regression ex: House Price Prediction

Assumptions of Linear Regression: (https://www.youtube.com/watch?v=EmSNAtcHLm8)

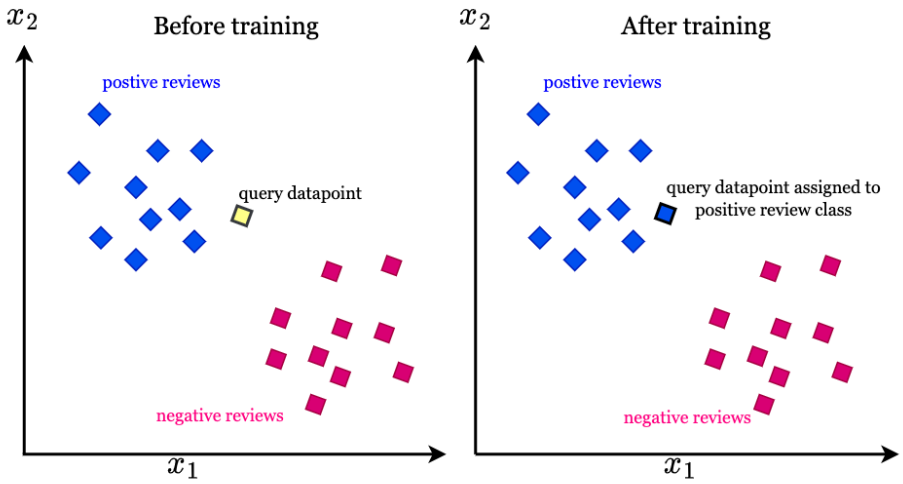
1. Linear Relationship between dependant and independent variables.
2. No Multicollinearity or No co-relation between independent variables.
3. Normal Residuals. (residual error should be in normal distribution I.e. mean 0 std dev. 1 if plotted graphically)
4. Homoscedasticity (Homo: same/uniform, scedasticity: Spread/scatter) residual spreads should be uniform.
5. No Auto Correlation of errors.



Limitations:

- Sensitive to outliers.
- Sensitive to missing values.

2. KNN: (Birds of a feather flock together)



Link: https://www.youtube.com/watch?v=IPqZKn_cMts&t=12s

Assumption: KNN Algorithm assumes that similar things exist in close proximity.

Use Case: Both Classification and Regression

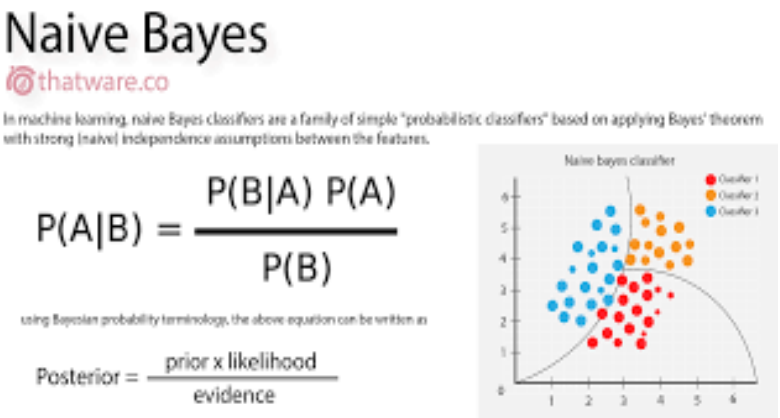
Explanation:

- Load data from dataset.
- Initialize 'k' to your chosen numbers of neighbors.
- For each sample of data, calculate the distance between the query (new unknown sample) and current example of data.
- Add distance and index of data in ordered collection
- Sort collection on basis of distances (ascending order)
- Pick 1st 'k' entries from sorted collection
- In classification case, get mode of 'k' labels
- In regression, get median of the labels.
- Distance can be calculated by:
    - Euclidian distance
    - Manhattan distance

Limitations:

- Sensitive to outliers.
- Sensitive to missing values
- Not good for huge dataset or a greater number of features.

3. Naïve Bayes: (https://www.youtube.com/watch?v=xXeoWE4KmmY)



Use Case: Classification

Assumption: This algorithm assumes that all features are independent and hence the word naïve.

Formula: P(A) = P(B). P(B|A) / P(A|B)

Proving Bayes theorem:

For **dependent** events, joint probability of A and B (such as deck of cards or picking green color marbles from box of yellow and green marbles)

P (A and B) = P(A). P(A|B)

Joint probability is commutative:

P (A and B) = P (B and A)

P(A). P(A|B) = P(B). P(B|A)

Thus, P(A) = P(B). P(B|A) / P(A|B)

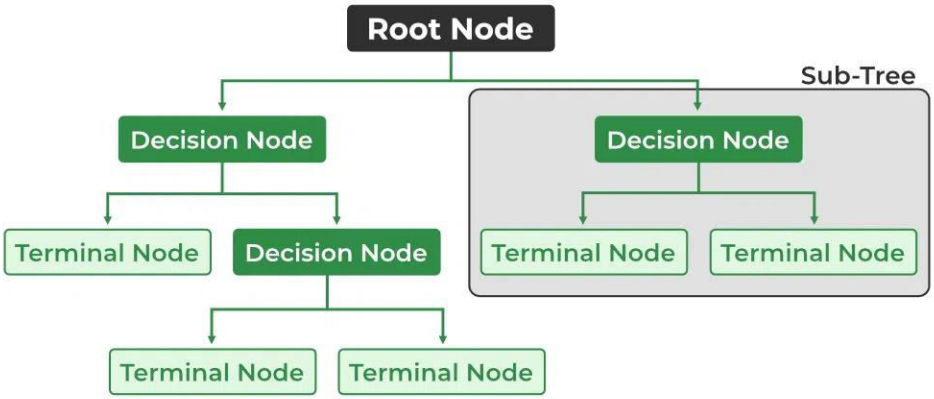
Pros:

- Used for high dimensional data such as text detection or spam detection
- Assumption is all features to be independent, makes this algorithm very fast.

Cons:

- Less accurate due to its assumption and not real case condition

4. Decision Trees: (https://youtu.be/ynTCUngbFHA?si=0ao-A_HSrQTEZVR3)



Use case: Both for classification and regression.

A decision tree is a flowchart-like tree structure where each internal node denotes the feature, branches denote the rules, and the leaf nodes denote the result of the algorithm.

A decision tree is prone to overfitting as it takes all the possibilities involved and decides till its leaf node.

Thus, sometimes to avoid overfitting pruning is performed.

Get to know about the terms like root node, internal nodes, leaf node, splitting.

Root node selection test is done on basis of highest **information gain, highest information gain feature is selected as root node.**

Purity check test:

Entropy: Entropy is the measure of the degree of randomness or uncertainty in the dataset.

Gini Impurity or index: Gini Impurity is a score that evaluates how accurate a split is among the classified groups. The Gini Impurity evaluates a score in the range between 0 and 1, where 0 is when all observations belong to one class, and 1 is a random distribution of the elements within classes.

Information Gain: Information gain is a measure of the reduction in impurity achieved by splitting a dataset on a particular feature in a decision tree

Pruning: The process of removing branches from the tree that do not provide any additional information or lead to overfitting.

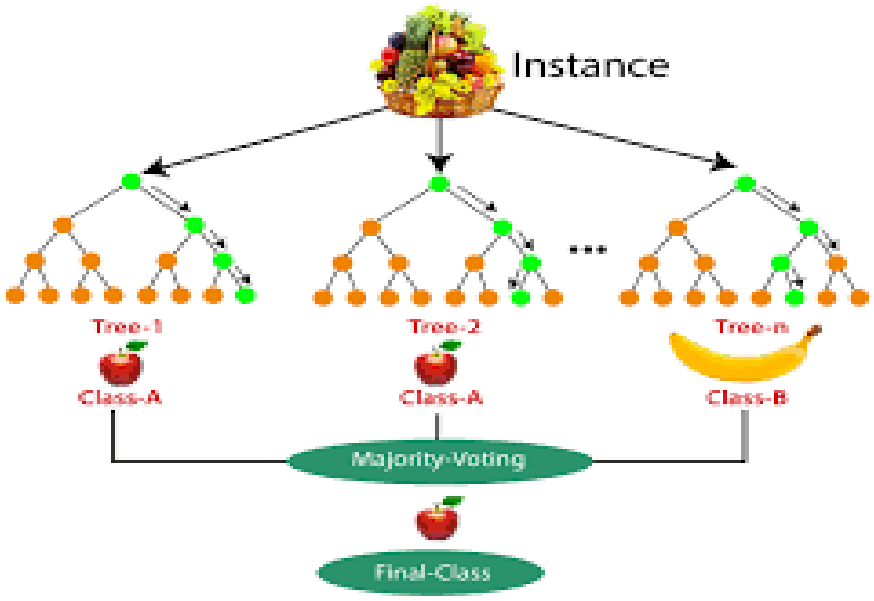
Pros:

- It is simple to understand as it follows the same process which a human follows while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

Cons:

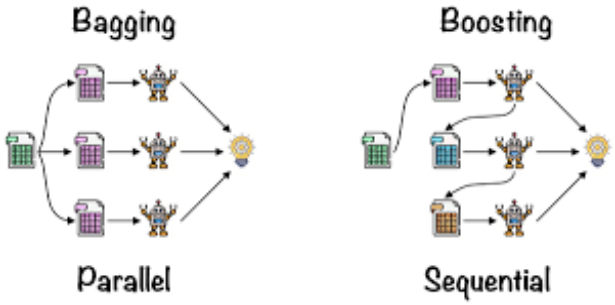
- The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the Random Forest algorithm.
- For more class labels, the computational complexity of the decision tree may increase.

5. Random Forest:



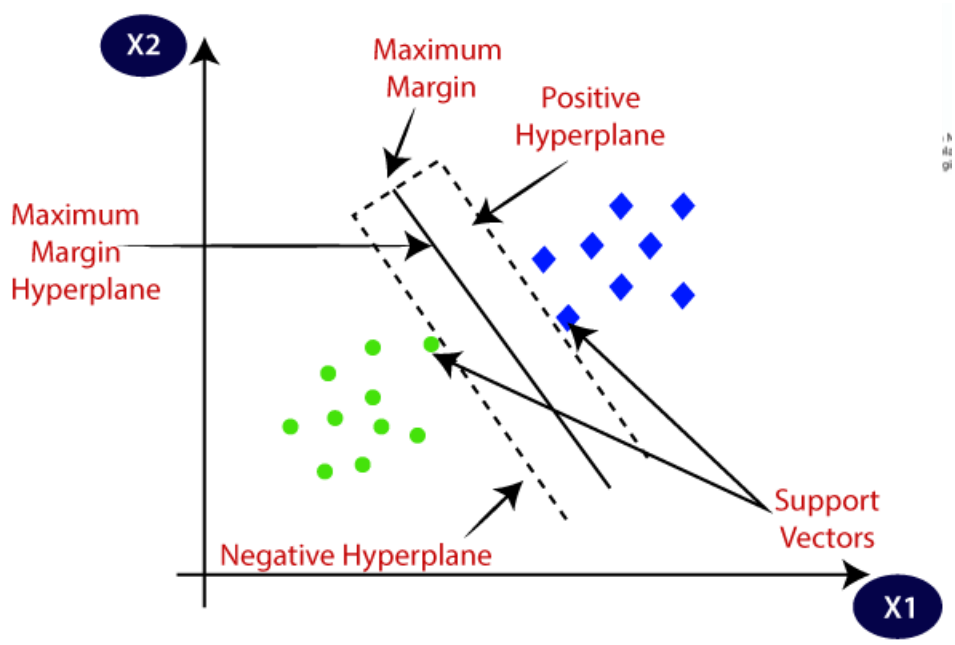
Use case: Both for classification and regression.

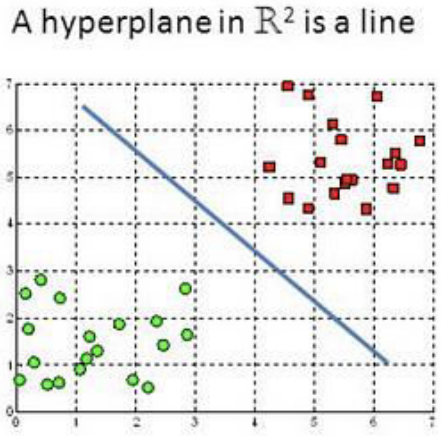
- Random Forest is an **ensemble** technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging.
- Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.
- Random Forest decreases overfitting (decreases variance) from decision trees as it gives subset of dataset to each decision tree (aka **bootstrapping**)
- In Random Forest classifier, median or mode is calculated to get output whereas for regressor, mean is calculated.
- Ensemble learning techniques can be categorized in three ways:
    - Bagging (Bootstrap Aggregating)
    - Boosting
    - Stacking (Stacked Generalization)

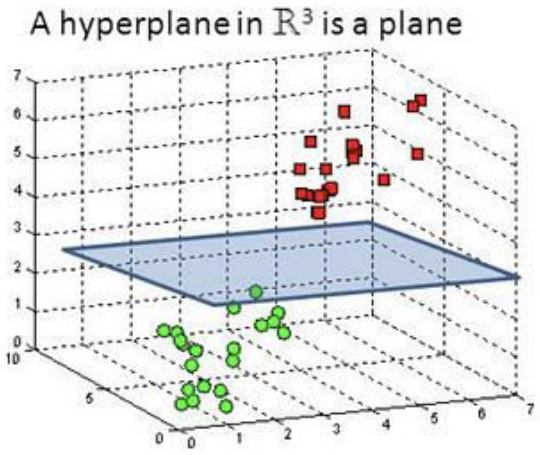


- **Bagging**: It is a homogeneous weak learners' model that learns from each other independently in parallel and combines them for determining the model average.
- **Boosting**: It is also a homogeneous weak learners' model but works differently from Bagging. In this model, learners learn sequentially and adaptively to improve model predictions of a learning algorithm.

6. Support Vector Machines (SVM):







- Here, we try to find a hyperplane that best separates the two classes.
- The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points.

- **Note:** Don't get confused between SVM and logistic regression. Both the algorithms try to find the best hyperplane, but the main difference is logistic regression is a

probabilistic approach whereas support vector machine is based on statistical approaches.

- SVM works best when the dataset is small and complex.
- Hyperplanes are decision boundaries that help classify the data points.
- The dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane.
- Hinge loss is used in the case of SVM.

Types of Support Vector Machine Algorithms

1. Linear SVM

- When the data is perfectly linearly separable only then we can use Linear SVM. Perfectly linearly separable means that the data points can be classified into 2 classes by using a single straight line (if 2D).

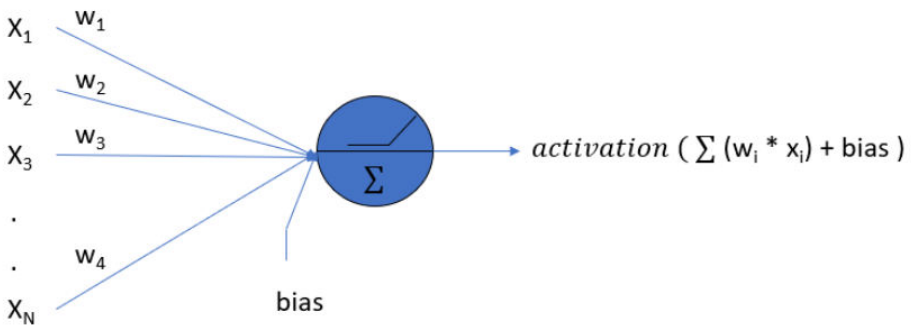
2. Non-Linear SVM

- When the data is not linearly separable then we can use Non-Linear SVM, which means when the data points cannot be separated into 2 classes by using a straight line (if 2D) then we use some advanced techniques like kernel tricks to classify them. In most real-world applications we do not find linearly separable datapoints hence we use kernel tricks to solve them.

Support Vectors: These are the points that are closest to the hyperplane. A separating line will be defined with the help of these data points.

Margin: it is the distance between the hyperplane and the observations closest to the hyperplane (support vectors). In SVM the large margin is considered a good margin. There are two types of margins **hard margin** and **soft margin.** I will talk more about these two in the later section.

7. Neural Networks:



A single neuron shown with X_i inputs with their respective weights W_i and a bias term and applied activation function

NN is a model with interconnected layers made of nodes.

Types of NN:

1. ANN:

ANN is also known as an artificial neural network. It is a feed-forward neural network because the inputs are sent in the forward direction

2. CNN:

Convolutional Neural Networks are mainly used for Image Data. It is used for Computer Vision.

3. RNN:

Recurrent Neural Networks. It is used to process and interpret time series data. In this type of model, the output from a processing node is fed back into nodes in the same or previous layers. The most well-known types of RNN are **LSTM** (Long Short-Term Memory) Networks.

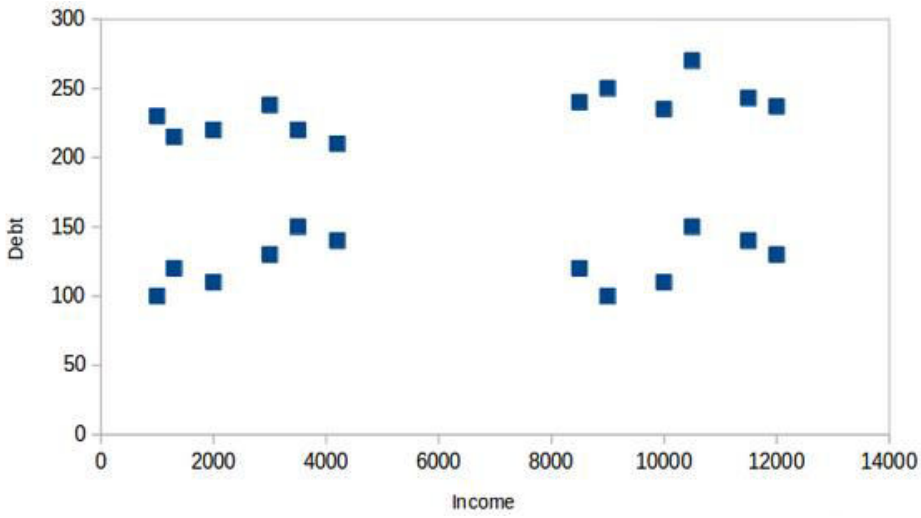
Unsupervised Algorithms

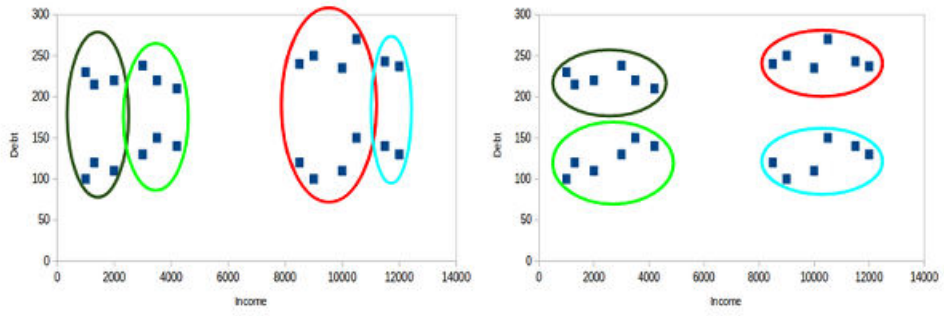
---

1. K-Means Clustering:

The process of creating groups is known as clustering.

We'll take the same bank as before, which wants to segment its customers. For simplicity purposes, let's say the bank only wants to use the income and debt to make the segmentation. They collected the customer data and used a **scatter plot** to visualize it:





Case - I Case - II

Properties of Clusters:

1. All the data points in a cluster should be similar to each other.
2. The data points from different clusters should be as different as possible.

Which of these cases do you think will give us the better clusters? If you look at case I:

Customers in the red and blue clusters are quite similar to each other. The top four points in the red cluster share similar properties to those of the blue cluster's top two customers. They have high incomes and high debt values. Here, we have clustered them differently. Whereas, if you look at case II:

Points in the red cluster completely differ from the customers in the blue cluster. All the customers in the red cluster have high income and high debt, while the customers in the blue cluster have high income and low debt value. Clearly, we have a better clustering of customers in this case.

Hence, data points from different clusters should be as different from each other as possible to have more meaningful clusters. The k-means algorithm uses an iterative approach to find the optimal cluster assignments by minimizing the sum of squared distances between data points and their assigned cluster centroid.

This algorithm is used in Recommendation systems or in Image segmentation where we try to club similar pixels in the image together.

2. Association Rule:

Please refer this algorithm from https://www.geeksforgeeks.org/association-rule/